RECEIVED
CENTRAL FAX CENTER

MAR 2 8 2006

AMENDMENTS TO THE SPECIFICATION

Please amend the specification as indicated by the following marked-up versions of the paragraphs that are to be amended in the specification:

[0036] In an embodiment of the invention, a programming interface is provided for allowing access to a repository without requiring the user or application interacting with the repository to understand the schema, syntax, or programming model of the repository. The interface, its environment and its function will be explained in greater detail with respect to the example of Figure 3, wherein the repository is an LDAP-compliant directory. In particular, a directory application 303 used by a user or other application to access the directory 311 directs high-level read, write, and search commands to a directory interface 305 via an interface thereto. These commands use the class definitions 306 generally used by the application 303 rather than the syntax and schema of the repository 311 itself. In particular, the access commands used by the directory application 303 do not use the schema or syntax of directory 311. Thus, the application 303 can use the same command structure and syntax for repositories of different types. Although the directory interface 305 is illustrated as a single entity, in an embodiment of the invention, the interface 305 comprises a plurality of separately callable entities, such as a read interface, search interface, and write interface.

[0042] As discussed herein, the interface 305 may use the class definitions 306 used by the application 303 in translating commands in either direction. Accordingly, the interface 305 preferably also supports an interface to the class definitions 306. In an embodiment of the invention, the directory interface 305 does not translate the command fully to one that embodies the lightweight directory access protocol. Rather the interface 305 translates the command to one that is appropriate for another API such as a system directory service API 307, which in turns fulfills the command through an LDAP API 309. The system directory service API 307 and LDAP API are familiar to those of skill in the art. Thus it will be appreciated that the interface between the directory interface 305 and the repository 311 may direct or may comprise additional APIs, and that the APIs may be changed appropriately to address different types of repositories, and that the APIs may be combined or further segmented without limitation.

[0051] The mapping of class structure classes to directory schema classes, and of class properties to schema class attributes, may be recorded or embodied in any appropriate form. For example, in an embodiment of the invention, the mappings are embodied in a listing that is accessible to the directory interface. However, in another embodiment of the invention, the mappings are embodied via metadata tagging within the class definition itself. An exemplary simplified class definition 306 and the tagging of the class definition 306 are illustrated in Figures 5A and 5B respectively. In particular, in Figure 5A, the class definition 501 appears with no metadata tagging.

[0052] The class is defined in Figure 5A as having a name 503 (PERSON), and as having three properties, "First Name" 505, "Last Name" 507, and "Address" 509.

However, in Figure 5B, the class definition 511 appears with metadata inserted. In particular, the class PERSON 513 has three sections of metadata 514, 515, 518 associating the three class properties, "First Name" 515, "Last Name" 517, and "Address" 519 with the LDAP attributes "Given Name", Surname," and "Location" respectively. Although each displayed property of the class PERSON 513 is associated with a metadata tag in Figure 5B, it should be appreciated that it is not necessary that each property be tagged. Furthermore, it should be appreciated that the directory schema classes that may be associated with the listed schema attributes need not be the same from one attribute to the next.